International Journal of Technology (2015) 2: 169-179
ISSN 2086-9614

IMPLEMENTATION AND PERFORMANCE ANALYSIS OF MOBILE HANDOFF PROCESS ON OPENFLOW-BASED WI-FI NETWORK

Ardiansyah[1*], Isyana Paramitha[1], Muhammad Salman[1], Deokjai Choi[2*]

[1]*Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia,*
Kampus UI Depok 16424, Indonesia
[2] *School of Electronics & Computer Engineering, Chonnam National University, South Korea*

(Received: February 2015 / Revised: April 2015 / Accepted: April 2015)

ABSTRACT

Research in communication networks has its limits due to the problems of supply frequency and equipment. To overcome this problem, open source can be the solution to build a helpful test bed for research or academic purposes. Open source networks can be developed using a Software Defined Network (SDN) which has been continuously developed due an enormous number of installed base equipment and protocols that are inflexible, predefined, and fixed since SDN offers a flexible, dynamic, and programmable functionality of network systems. By using OpenFlow as its protocol, we can program the network flow in a flow table on different switches and routers. This research approches an OpenFlow-based Wi-Fi environment using an OpenFlow-based Access Point (OFAP) and an OpenFlow controller. Each OFAP is deployed in two different rooms and several experiments were performed to evaluate handoff delay. The result of these experiments shows that an OpenFlow-based network delivers a more stable process than a traditional network because of the installed flows that are given to each packet. However, the discovered value needs to be examined further due to a better mechanism for installed flows. The handoff delay between OFAPs is 24% faster than the handoff delay between a traditional AP with an average of 79.9 miliseconds. By using this system, we believe it could deliver a high performance network and an increased reliability for real-time traffic over WLAN, by reducing the handoff delay compared to a classical Wi-Fi environment.

Keywords: Access point; Delay; Handoff; OpenFlow; Wi-Fi

1. INTRODUCTION

The development of network and communication devices evolve correspondingly with Internet access as the Internet is the most widely used communication media between people. Specifically, mobile technology has the fastest development in the case of communication devices. There are an estimated 6.8 billion people in the world, where 4 billion of them have their own mobile phones (Afshar, 2014). One-third of the volume of mobile devices belongs to smartphones and media tablets. This development affects the network technology as well, such as Wi-Fi, which is synonymous with Wireless Local Area Network. (WLAN). Its speed has grown from 54 Mbps on 802.11a, 450 Mbps on 802.11n, up to 1.3 Gbps on 802.11ac in 2012 (IEEE, 2013). It is predicted that in 2017, 71% of the communication devices using mobile phone protocols will use Wi-Fi to connect to the Internet (Mishra, Shin & Arbaugh, 2003).

---

[*] Corresponding author's email: ardisragen@eng.ui.ac.id, Tel. +62-21-7270078, Fax. +62-21-7270077
Permalink/DOI: http://dx.doi.org/10.14716/ijtech.v6i2.1004

But one Wi-Fi network cannot covered one big area, so where a user has to move from one place to another, which may result in signal degradation, the connection will be moved to another Wi-Fi network by a process called, ‘handoff’. In this process, the user will be moved electronically to another access point (AP). By the time this process takes place, the user will no longer be connected to the Internet. Connections would occur between the mobile node and the new AP by exchanging an 802.11 management frame to build a new session. But to build the new session, there are some phases where each phase will need time, which produces a handoff delay. As the development goes on, this handoff delay has become a concern in research areas. There are some schemes developed as the solution from minimizing handoff delays by using a fast-handoff protocol (Amir et al., 2006; Zhang et al., 2007; Teng et al, 2009) until the priority-aware function manages to support the Quality of Service (QoS) in the Wi-FI handoff (Tri & Kim, 2014).

Network technology also aims to build an open network system using Software Defined Networking (SDN) architecture and OpenFlow as its most common protocol, since there are some limitations due to frequency and equpment supply. The general idea of the SDN is to separate the network control function and the data plane in order to forward the information packets, so the network flow will be easily controlled by user. Using this architecture will unleash the limitations on the current Internet use by substituting the fixed and old-fashioned existing network. Related to the facts in the previous paragraph, it is imperative to build the open source technology in order to establish the future Internet infrastructure by developing the new open mobile wireless network structure for the mobile communications market. Before it will be used publicly, it has to be tested amongst researchers and this paper will evaluate a handoff project using OpenFlow-based Wi-Fi on a mobile device laid horizontally between two OpenFlow-based APs. The network flow will be modified based on the topology, as well as the maximum time before the connection ended. The aim of this research is to find a new solution of the handoff project as well as the beginning of implementing open network technology before it will be widely deployed,.presumably on a commercial basis. The rest of this article is organized as follows: an overview of the handoff process in the IEEE 802.11 network, followed by the testbed topology, evaluation and data analysis from the captured packets, and finalized by concluding remarks.

2. HANDOFF PROCESS IN IEEE 802.11

The handoff process occurs when a user moves from one access point (AP) to another because of signal degradation, so time is necessary to build a new session with the new AP. Shin et al., 2004 state that there are two phases in the handoff process followed by three kinds of handoff delay.

2.1. Discovery

The Signal to Noise Ratio (SNR) is degraded when there is mobility of mobile devices referred to a Mobile Node, (MN), which triggers the handoff process. Before it closes the connection to an AP (AP1) and moves to new AP (AP2), the MN has to find the potential AP2. It will be done in the Media Access Control (MAC) layer using a scanning function.

There two kinds of scanning on the Wi-Fi network: active and passive. Passive scanning means MN will be paying attention to “listen” to the Beacon Frame, one of the management frames in the IEEE 802.111 WLAN. This frame will provide timing information and statements to help the MN to consider which AP to connect with. The current IEEE 802.11 standard covers multiple channel scanning. Specifically, 802.11b and 802.11g standards operate on 2.4 GHz broadband using 11 channels from the 14 provided. 802.11a operates on a 5 GHz broadband which has 32 channels. When the passive mode is working, a MN will listen to each channel to find the new AP so this process produces quite a significant delay.

On the other hand, an active scanning mode involves a frame transmission activity which can be described step-by-step as follows:

1. A normal channel procedure, where the CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) is activated, to take over the wireless media control.
2. MN transmits a probe request frame.
3. Probe timer starts.
4. MN waits for a response
5. If there is no response in the MinChannelTime time, it will scan the next channel.
6. If there is more than one response received, MN will stop the incoming frame on the MaxChannelTime and then process all the received information.
7. These steps will be re-done for each channel.

2.2. Reassocation

This procedure covers authentication and reassociation delay towards the AP2 and the information transfer process from AP1. Authentication is a process where the AP chooses to accept or deny a connection request from the MN. The MN will start the authentication process by sending the authentication process frame to the AP which has the MN's identity information. The AP will respond by sending an authentication response frame to MN which then decides whether the MN will be accepted or denied. After a successful authentication, MN will send the reassociation request frame and respond with a reassociation response frame which contains another 'accept' or 'deny' command.

2.3. Handoff Delay

The total delay time is divided into three kinds of delays:

2.3.1. Probe delay

Probe delay depends on which mode scan has been used; and which system is in use: active or passive. The average of probe delay on the passive scanning mode can be represented by the function of the beacon frame interval and the number of available channels. For example, if the beacon interval is 100msec, the average probe delay for IEEE 802.11b with 11 channels is 1100 msec and 802.11b is 3200 msec. Plus the switching delay is estimated around 40-150 usec. On active scanning mode, probe delay can be determined from the value of MinChannelTime and MaxChannelTime based on the device. Active scanning procedures require the MN to scan all available channels. We can determine the total delay by this function:

$$N \times MinChannelTime \leq T_A \leq N \times MaxChannelTime \tag{1}$$

where N is the total available channels.

Based on the above functions, simply put, the best way to reduce the probe delay is by reducing the number of scanned channels. Scanning can be done by choosing some channels, based on policy that is then executed by the protocol administrator. Another method is by improving the MinChannelTime and the MaxChannelTime.

2.3.2. Authentication delay

Delay occurs when there is a frame exchange process. There are two authentication modes: open-system authentication where the AP will accept any MN without authentication or by using the MAC address filtering (which is not included in the 802.11 standard); and Shared Key Authentication (SKA) using the Wired Equivalent Privacy (WEP) (Corvaja, et.al., 2004), which needs the AP and the MN to implement it. This authentication exchanges four kinds of messages as follows:

1. Challenge-Request message from the MN to the AP asking for authentication
2. Challenge-Response message contains a random number from the AP to the MN
3. The MN will sign the random number using the WEP as the distributed secret-key between the AP and the MN through the safe channel before it is used. Then it sends a Response message to the AP
4. The AP verifies the signed random number with the valid key by counting it and comparing it with the received value. After the key is verified, the AP will authenticate the MN by sending an Approval message

The total authentication delay time depends on the number of sent messages between the AP and the MN, so the Shared Key Authentication (SKA) will take a longer time when compared to an Open System Authentication (OSA) mode.

2.3.3. Reassociation delay

Reassociation is a process to move a connection or association from one AP to another within an Extended Service Set (ESS). An ESS contains some interconnections of the Basic Service Set (BSS) where BSS is a set of AP services. Similar with authentication delay, the value of Reassociation Delay is based on the Reassociation Request Frame exchange.

After the authentication process is done, the MN will send the Reassociation-Request frame to the AP and receives Reassociation-Response frame and finish the handoff process. It similar with the authentication process, but it is not on the backbone network where the APs will communicate reciprocally to send the frames which are related to the reassociation process.

3. EXPERIMENTAL SETUP

The testbed consisted of one server, two OpenFlow-based Access Points (APs) and a Mobile Node (MN). The server was configured as the access controller using OpenDayLight.The router used Quagga, and the video streamer. AP was configured as an OpenFlow-based AP using Pantou. An Android-based application, tPacketCapture, was also installed in the MN to capture the packet flow when the process was going on. Two Aps were located in different areas as shown in Figure 1. This arangement means that there was no signal interference between the two APs.

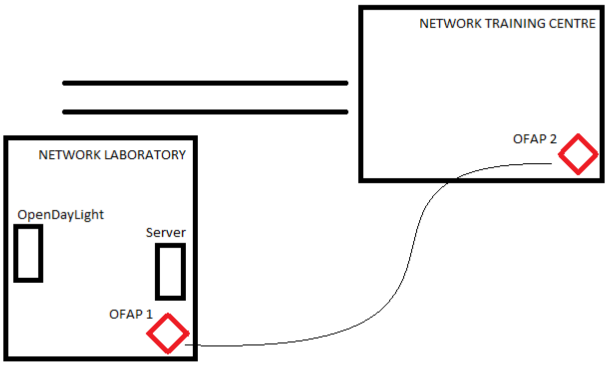
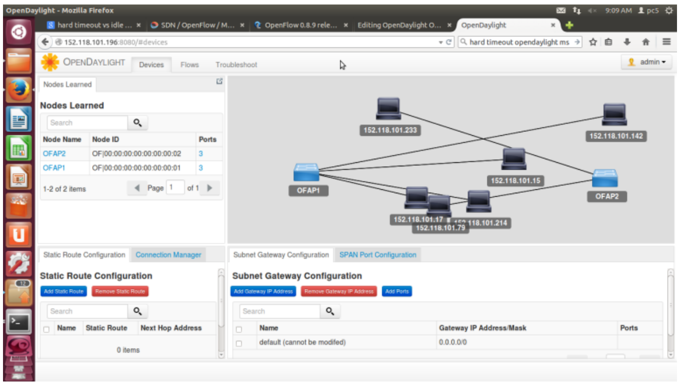


Figure 1 Map where the APs were installed

The server IP address was static: 152.118.101.196. OpenDayLight had to be activated before it could be used as the controller, it was also the same with Quagga. The Quagga configuration consisted of several daemons in which each represented a supported routing protocol. In this research, OSPF and Zebra as the routing kernel were activated. VLC was used to stream the video so the MN could be streaming while it captured the packet flow at the same time.

The APs were flashed with OpenWrt and OpenFlow1.0 image based on an Attitude adjustment regarding the device's brand and type. (This research used Buffalo WZR-HP-G450H). OpenFlow was configured by putting the controller's IP address (152.118.101.196) on the 'ofctl' parameter. The testbed connection could be checked through OpenDayLight as depicted in Figure 2.



Nodes Learned

Node Name	Node ID	Ports
OFAP2	OF|00:00:00:00:00:00:00:02	3
OFAP1	OF|00:00:00:00:00:00:00:01	3

1-2 of 2 items Page 1 of 1

Static Route Configuration

☐	Name	Static Route	Next Hop Address

0 items

Subnet Gateway Configuration

☐	Name	Gateway IP Address/Mask	Ports
☐	default (cannot be modifed)	0.0.0.0/0	

Figure 2 OpenDayLight screenshot

OpenDayLight detects OpenFlow-based switch (in this case was the AP) and it was used to configure the network traffic by installing the network flows. In this research, the flow was configured by describing each flow's input and output port. The communication flow was set as in Figure 3.

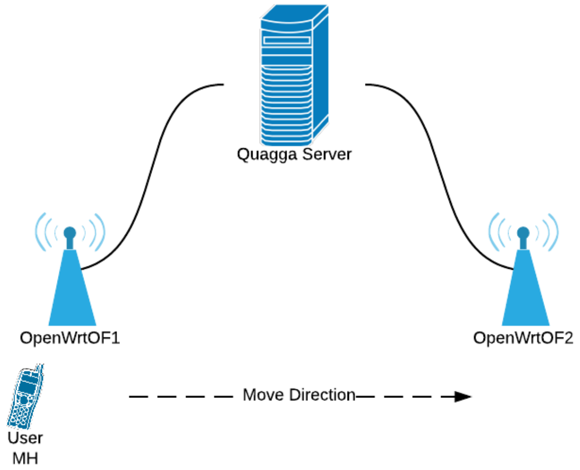


Figure 3 Communication flow

The MN is moved from one location to another. In the first location it was connected with OFAP1 then it was moved until OFAP1's signal degraded and it was changed to OFAP2 which had a better signal which automatically started the handoff process.

4. RESULTS AND DISCUSSION

The recorded data was taken using two devices, a mobile device and a laptop. tPacketCapture was used on the mobile device, and Wireshark was used on the laptop. It was done considering the deficiency of tPacketCapture which was only capturing packets on Layer 3, while the handoff process used the frame on Layer 2 to communicate. The data was taken 28 times using a mobile device with a chronology as reported in Table 1.

Table 1 Experiment's chronology

Data-	Explanation
1	Handoff process was evaluated by moving the mobile device from OFAP1 to OFAP2 normally, with TCP and DNS packets recorded clearly.
2	Handoff process was evaluated by moving the mobile device from OFAP1 to OFAP2 normally, with TCP and DNS packets recorded clearly.
3	Handoff process was evaluated by moving the device from OFAP1 to OFAP2 normally, the recorded packet did not indicate that the device had been moved to the new OFAP.
4	Handoff process was evaluated by moving the device from OFAP1 to OFAP2 normally, the recorded packet did not indicate that the device had been moved to the new OFAP.
5	The recorded data did not comply with the set parameter
6	The recorded data did not comply with the set parameter
7	The recorded data did not comply with the set parameter
8	Handoff process was evaluated by moving the mobile device from OFAP1 to OFAP2 normally, with TCP and DNS packets recorded clearly.
9	Handoff process was evaluated by moving the device from OFAP1 to OFAP2 normally, the recorded packet did not indicate that the device had been moved to the new OFAP.
10	Handoff process was evaluated by moving the device from OFAP1 to OFAP2 normally, the recorded packet did not indicate that the device had been moved to the new OFAP.
11	Handoff process was evaluated by moving the mobile device from OFAP2 to OFAP1 normally, with TCP and DNS packets recorded clearly.
12	Handoff process was evaluated by moving the device from OFAP1 to OFAP2 normally, the recorded packet did not indicate that the device had been moved to the new OFAP.
13	Handoff process was evaluated by moving the mobile device from OFAP2 to OFAP1 normally, with TCP and DNS packets recorded clearly.
14	Handoff process was evaluated by moving the device from OFAP1 to OFAP2 normally, the recorded packet did not indicate that the device had been moved to the new OFAP.
15	The recorded data did not comply with the set parameter
16	The recorded data did not comply with the set parameter
17	Handoff process was evaluated by moving the device from OFAP1 to OFAP2 normally, the recorded packet did not indicate that the device had been moved to the new OFAP.
18	Handoff process was evaluated by moving the device from OFAP1 to OFAP2 normally, the recorded packet did not indicate that the device had been moved to the new OFAP..
19	Handoff process was evaluated by moving the device from OFAP1 to OFAP2 normally, the recorded packet did not indicate that the device had been moved to the new OFAP.
20	Handoff process was evaluated by moving the mobile device from OFAP1 to OFAP2 normally, with TCP and DNS packets recorded clearly.
21	Handoff process was evaluated by moving the mobile device from OFAP1 to OFAP2 normally, with TCP and DNS packets recorded clearly. But the last TCP packet was disconnected between the device and the server (the mobile device's service connection).
22	Handoff process was evaluated by moving the mobile device from OFAP1 to OFAP2 normally, with TCP and DNS packets recorded clearly.
23	Handoff process was evaluated by moving the mobile device from OFAP2 to OFAP1 normally, with TCP and DNS packets recorded clearly.
24	Handoff process was evaluated by moving the mobile device from OFAP1 to OFAP2 normally, with TCP and DNS packets recorded clearly. But the last TCP packet was disconnected between the device and the server (the mobile device's service connection).
25	Handoff process was evaluated by moving the mobile device from OFAP2 to OFAP1 normally, with TCP and DNS packets recorded clearly.
26	Handoff process was evaluated by moving the mobile device from OFAP1 to OFAP2 normally, with TCP and DNS packets recorded clearly.
27	Handoff process was evaluated by moving the mobile device from OFAP1 to OFAP2 normally, with TCP and DNS packets recorded clearly.
28	Handoff process was evaluated by moving the mobile device from OFAP1 to OFAP2 normally, with TCP and DNS packets recorded clearly.

As stated above, five out of the 28 data instances collected could not be used since the parameters were not fulfilled. The conditions indicated that the streaming process was not going well and there was no handoff process because of the *testbed* instability. From 23 data that fulfilled those parameters, nine were not counted because the recorded packets were not providing the shifting process between the two OFAPs clearly. For instance, in the third data, the shifting process happened from OFAP1 to OFAP2, but there was no recorded DNS packet addressed to 192.168.2.1 (where the OFAP2 was designated as the new OFAP). One possible reason for this issue was caused by the packet capturer application which used a VPN. When the device was connected to the new OFAP and it received a new IP address, it could not be seen since it was recorded as a private IP address. Delay calculations on those nine data could be assumed, but it was not done in this research. The final count was a delayed calculation based on the 14 data which had fulfilled the parameters taken from the mobile device, which was analyzed using Wireshark.

Based on the need for VoIP communication or generally real-time multimedia communication, 100 millisecond (ms) is required before the connection is considered lost (Corvaja et al., 2004). The expected delay time resulted in a reading of less than 100 ms. The data analysis was done by counting the time difference (Δt) between the last data sent or received by the TCP packet to and from server and the first DNS packet addressed to the new OFAP. For every instance of data taken, except for the 21st and 24th data in which the last TCP packet was counted as a connection with another receiver, the findings were based on running the mobile device's service. As explained before, there are only 14 data which indicated the DNS packet clearly. The result is as shown in Figure 4.

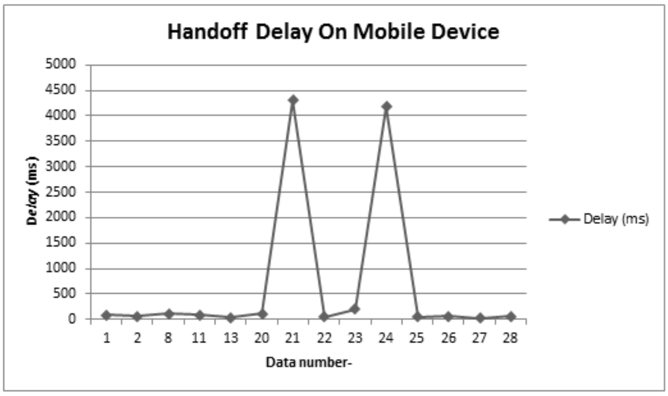


Figure 4 Handoff delay value for each data taken with mobile device

The 21st and 24th data are anomalies because of the running application service. The average delay time is 79.9 millisecond.

Another test was taken four times using a laptop as the mobile device which resulted in the graph shown in Figure 5.

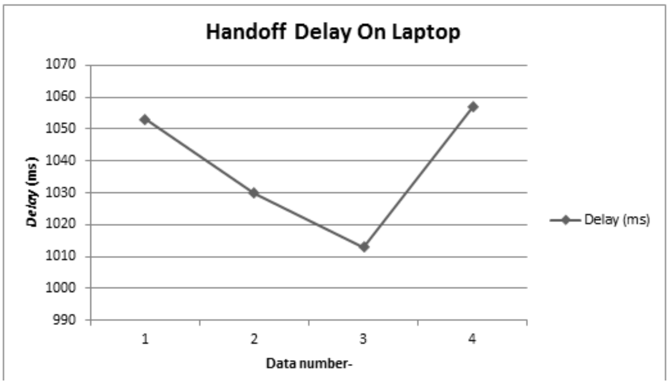


Figure 5 Handoff delay value for each data taken with laptop,
The average delay using laptop is 1038.25 millisecond

The first result is based on 14 data minus 2 anomaly data that have fulfilled the target (<100 ms). Additionally, the fact that the recorded PDUs are layers of 3 OSI-layered Protocol Data Units (PDUs), the result could be assumed as a “gross” result. With further selection of packets and frame combinations (Layer 2 OSI-layered PDUs), the outcome can be reduced. But further tests need to be performed, since 14 data are not being considered sufficiently to indicate that this method could be used as the solution to reduce the delay handoff process for real time multimedia communication.

The delay value from the second graph was counted by subtracting the time between last data sent or received by the TCP packet to and from server and the first sent to the Dynamic Host Configuration Protocol (DHCP) frame. Since the default handoff process is provided by the DHCP settings, the value is quite high (Corvaja, et.al., 2004).This data can be analysed because the laptop uses a better Network Interface Card (NIC) than the NIC on the mobile device. There were a lot more recorded packets and frames which resulted in a longer delay time. But this test was done to portray frame communication in the handoff process which cannot be done by using the *tPacketCapture* application.

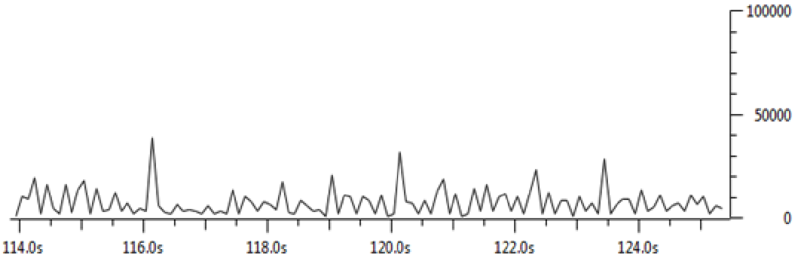


Figure 6 Recorded time (x-axis) versus packet size (y-axis) graph

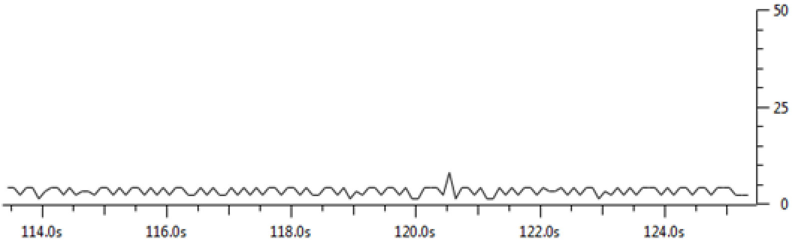


Figure 7 Recorded time (x-axis) versus the amount of packet (y-axis) graph

The above Figures 6 and 7 graphs indicated that the packet size (bytes) and the number of packets were hardly interrupted and these were never valued in 0. It shows that the quality of communicated data was better and it could be assumed that the shifting between two OFAPs are more stabile because the OpenFlow connection stability was better than the traditional connection.

To calculate the total delay time or the handoff process latency, unfortunately there was still no one general standarization. This research used a raw handoff latency calculation scheme. This scheme calculated the time interval from the requested first probe from client/user until the re-association response from the new AP was measured, with an added probe_delay_time as formulated below:

$$\text{Rhl} = \text{ProbeDelay} + (\text{ReAssResponse} - \text{FirstProbeReq}) \qquad (2)$$

Where Rhl is the total Raw Handoff Latency, the ReAssResponse is the time when the first re-association response is received. The FirstProbeReq for the First Probe Request message is sent from client and the ProbeDelay (prob_delay_time) is the time limit which is given as the maximum limit for the discovery process in the discovery phase. For this function it was assumed that it had already been implemented in the Δt, as a result of the experiment, without any clear delay segmentation, due in part to the lack of application data. The Rhl value from the experiment using mobile device was 79.9 ms and using a laptop was 1038.25 ms, respectively.

Table 2 Handoff delay measurement

***Delay Handoff* (ms)**	**The Amount of Data**
50-59	7
60-69	7
70-79	9
80-89	4
90-99	9
100-109	20
110-119	26
120-129	15
130-139	1
140-149	1
150-159	1
Average Delay Time	**100,6 ms**

The Delay Time result during the handoff process in the experiment for the traditional Wi-Fi network is depicted in Table II (Corvaja et al., 2004).Those results were taken using a *testbed* where two APs are placed within a minimal overlap signal. This shifting data that occured still could be applied as a fast handoff. Therefore, our research with the Rhl value indicated a better value than the Delay Result.Yet, this data still need to be re-evaluated so the findings could legitimate the proposed theory. This research resulted in a high possibility that the data can be used as a solution to minimize the delay time in real-time multimedia communication. Beside the minimum delay, OpenFlow also has the potential to be implemented widely with higher flexibility, control, and lower cost implications when compared to the traditional network.

5. CONCLUSION

The results from this expriment are concluded as follows:

1. The average of handoff delay process on an OpenFlow-based Wi-Fi network using fast handoff scheme is 79.9 millisecond, when using a mobile device installed by a packet capturer application as the measurement tool.
2. The average handoff delay process on OpenFlow-based Wi-Fi network using a fast-handoff scheme is 1038.25 millisecond because of a differentiation in stack settings between the packet capturer applications used on both a mobile device and a laptop. It caused the number and variance of packet and frame data to increase and this occurence impacted the delay value.
3. The resultant data analysis cannot completely conclude that an OpenFlow-based network could be used as the solution for handoff delay in real-time multimedia communications. The simplistic built topology and the formulated data flows from controller have not supported the aim of this research to minimize the delay in the handoff process.
4. On the other hand, the results show that this architecture has the potential to be a solution of handoff delay in real-time multimedia communications.
5. Besides a solution for the delayed signals in the wireless network, OpenFlow implementation in the Wi-Fi network could be developed in increasing the network flexibility and reducing the cost, as well as changing the traditional handoff process which from client initiated to network initiated signals

Related to the several shortcomings as indicated above, if this research could be re-evaluated to obtain stronger results as a proof for a solution to minimise the handoff delay time using SDN architecture, then there would be some relevant points for recommendation and consideration as indicated below.

1. Network Without Proxy: Since this research is done in a campus environment equipped with a proxy-network, it needs some IP addresses included in DMZ network segment.
2. Topology Re-designing. The built topology is too simple. Between the controller and the OFAPs there needs to be an additional device, such as an OpenFlow-based switch so the control to both OFAPs will be more stable.
3. Better Formulation of Flows. Adding signal flows by using a OpenDayLight controller is the easiest and simplest method, as well as an additional user interface feature which would be more effective, but the needed parameters to give the time limit for probe delay (in millisecond) still cannot be done. Alterations can be instigated by using another controller (ProGFE, for instance) or by operating the OpenDayLight through a command line so the parameter configuration can be accomplished effectively.

6. ACKNOWLEDGEMENT

This research was managed by the international research collaboration between Network Laboratory, Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia and Advanced Network System Laboratory, School of Electronics & Computer Engineering, Chonnam National University, South Korea supported by the Korea National Information Society Agency (NIA).

7. REFERENCES

Afshar, V., 2014. 50 Incredible WiFi Tech Statistics that Businesses Must Know. Available online at http://www.huffingtonpost.com/vala-afshar/50-incredible-wifi-tech-s_b_4775837.html

Amir, Y., Danilov, C., Hilsdale, M., Musaliou-Elefteri, R., Rivera, N., 2006. Fast Handoff for Seamless Wireless Mesh Networks. *MobiSys'06*, June 19-22, 2006, Uppsala, Sweden

Corvaja, R., Zanella, A., Dossi, M., Tontoli, A., Zennaro, P., 2004. Experimental of the Handover Procedure in a WiFi Network. WPMC04

Hiep, T.N.T., Kyungbaek, K., 2014. Supporting WiFi QoS with Priority-aware WiFi Handoff. *Advanced Science and Technology Letters*, Volume 46 (Mobile and Wireless 2014), pp.15–19

IEEE 802.11ac Standard, 2013. Wireless Lan Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High Throughput (HT) PHY specifications

Mishra, A., Shin, M., Arbaugh, W., 2003. An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process. *ACM SIGCOMM Comp. Commun. Review*, Volume 33(2), April 2003, pp. 93–102

Shin, S., et al., 2004. Reducing MAC Layer Handoff Latency in IEEE 802.11 Wireless LANs. *Proc. ACM MobiWac 2004*

Teng, J., Xu, C., Jia, W., Xuan, D., 2009. D-Scan: Enabling Fast and Smooth Handoffs in AP-dense 802.11 Wireless Networks. *IEEE INFOCOM*, Rio de Janeiro, Brazil.

Zhang, Y., Liu, Y., Xia, Y., Huang, Q., 2007. LeapFrog: Fast, Timely WiFi Handoff. *Global Telecommunications Conference*, Washington, DC, USA